StreamStor Serial FPDP Daughterboard

User Manual

Copyright and Trademarks

The information in this document is subject to change without notice.

This document contains proprietary information that is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced, or translated to another language without the prior written consent of Conduant Corporation.

Printed in the United States.

© 2008 Conduant Corporation. All rights reserved.

StreamStor is a trademark of Conduant Corporation.

All other trademarks are the property of their respective owners.

Version 8.10

Publication date: March 13, 2008

TABLE OF CONTENTS

LICENSE AGREEMENT AND LIMITED WARRANTY 4

ABOUT THIS MANUAL 6

OVERVIEW 7

OPTICAL INTERFACE 7
OPTICAL PORT DESCRIPTION 9

PROGRAMMING 10

CONFIGURING SFPDP WITH XLRSETMODE 10
Forking and PassThru 10
CONFIGURING SFPDP WITH XLRSETDBMODE 11
SETTING UP CHANNELS 12
Single Channel Operation 13
Multi-Channel Operation 14
SETTING OPTIONS WITH XLRSETOPTION 15
SERIAL FPDP DAUGHTERBOARD LED DESCRIPTIONS 16
C CODING EXAMPLE 17

TECHNICAL SUPPORT 19

License Agreement and Limited Warranty

IMPORTANT: CAREFULLY READ THE TERMS AND CONDITIONS OF THIS AGREEMENT BEFORE USING THE PRODUCT. By installing or otherwise using the StreamStor Product, you agree to be bound by the terms of this Agreement. If you do not agree to the terms of this Agreement, do not install or use the StreamStor Product and return it to Conduant Corporation.

GRANT OF LICENSE. In consideration for your purchase of the StreamStor Product, Conduant Corporation hereby grants you a limited, non-exclusive, revocable license to use the software and firmware which controls the StreamStor Product (hereinafter the "Software") solely as part of and in connection with your use of the StreamStor Product. If you are authorized to resell the StreamStor Product, Conduant Corporation hereby grants you a limited non-exclusive license to transfer the Software only in conjunction with a sale or transfer by you of the StreamStor Product controlled by the Software, provided you retain no copies of the Software and the recipient agrees to be bound by the terms of this Agreement and you comply with the RESALE provision herein.

NO REVERSE ENGINEERING. You may not cause or permit, and must take all appropriate and reasonable steps necessary to prevent, the reverse engineering, decompilation, reverse assembly, modification, reconfiguration or creation of derivative works of the Software, in whole or in part.

OWNERSHIP. The Software is a proprietary product of Conduant Corporation which retains all title, rights and interest in and to the Software, including, but not limited to, all copyrights, trademarks, trade secrets, know-how and other proprietary information included or embodied in the Software. The Software is protected by national copyright laws and international copyright treaties.

TERM. This Agreement is effective from the date of receipt of the StreamStor Product and the Software. This Agreement will terminate automatically at any time, without prior notice to you, if you fail to comply with any of the provisions hereunder. Upon termination of this Agreement for any reason, you must return the StreamStor Product and Software in your possession or control to Conduant Corporation.

LIMITED WARRANTY. This Limited Warranty is void if failure of the StreamStor Product or the Software is due to accident, abuse or misuse.

Hardware: Conduant's terms of warranty on all manufactured products is one year from the date of shipment from our offices. After the warranty period, product support and repairs are available on a fee paid basis. Warranty on all third party materials sold through Conduant, such as chassis, disk drives, PCs, bus extenders, and drive carriers, is passed through with the original manufacturer's warranty. Conduant will provide no charge service for 90 days to replace or handle repair returns on third party materials. Any charges imposed by the original manufacturer will be passed through to the customer. After 90 days, Conduant will handle returns on third party material on a time and materials basis.

Software: The warranty on all software products is 90 days from the date of shipment from Conduant's offices. After 90 days, Conduant will provide product support and upgrades on a fee paid basis. Warranties on all third party software are passed through with the original manufacturer's warranty. Conduant will provide no charge service for 90 days to replace or handle repair returns on third party software. Any charges imposed by the manufacturer will be passed through to the customer.

DISCLAIMER OF WARRANTIES. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CONDUANT CORPORATION DISCLAIMS ALL OTHER WARRANTIES AND CONDITIONS, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND NONINFRINGEMENT, WITH REGARD TO THE STREAMSTOR PRODUCT AND THE SOFTWARE.

SOLE REMEDIES. If the StreamStor Product or the Software do not meet Conduant Corporation's Limited Warranty and you return the StreamStor Product and the Software to Conduant Corporation, Conduant Corporation's entire liability and your exclusive remedy shall be at Conduant Corporation 's option, either (a) return of the price paid, if any, or (b) repair or replacement of the StreamStor Product or the Software. Any replacement Product or Software will be warranted for the remainder of the original warranty period.

LIMITATION OF LIABILITIES. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL CONDUANT CORPORATION BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR ANY OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OF OR INABILITY TO USE THE STREAMSTOR PRODUCT AND THE SOFTWARE. IN ANY CASE, CONDUANT CORPORATION'S ENTIRE LIABILITY UNDER ANY PROVISION OF THIS AGREEMENT SHALL BE LIMITED TO THE AMOUNT ACTUALLY PAID BY YOU FOR THE STREAMSTOR PRODUCT AND THE SOFTWARE. BECAUSE SOME STATES AND JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY, THE ABOVE LIMITATION MAY NOT APPLY TO YOU.

RESALE. If you are authorized to resell the StreamStor Product, you must distribute the StreamStor Product only in conjunction with and as part of your product that is designed, developed and tested to operate with and add significant functionality to the StreamStor Product; you may not permit further distribution or transfer of the StreamStor Product by your end-user customer; you must agree to indemnify, hold harmless and defend Conduant Corporation from and against any claims or lawsuits, including attorneys' fees, that arise or result from the use or distribution of your product; and you may not use Conduant Corporation's name, logos or trademarks to market your product without the prior written consent of Conduant Corporation.

ENTIRE AGREEMENT; SEVERABILITY. This Agreement constitutes the complete and exclusive agreement between you and Conduant Corporation with respect to the subject matter hereof and supersedes all prior written or oral agreements, understandings or communications. If any provision of this Agreement is deemed invalid under any applicable law, it shall be deemed modified or omitted to the extent necessary to comply with such law and the remainder of this Agreement shall remain in full force and effect.

GOVERNING LAW. This Agreement is governed by the laws of the State of Colorado, without giving effect to the choice of law provisions therein. By accepting this Agreement, you hereby consent to the exclusive jurisdiction of the state and federal courts sitting in the State of Colorado.

About This Manual

This manual is intended to serve the following purposes:

- to provide an overview of the StreamStor Serial FPDP daughterboard;
- to act as a reference for the operator; and
- to provide guidance on software capabilities and choices.

It is suggested that you periodically check the Conduant web site for the most recent software updates, application notes, and technical bulletins.

If you are unable to locate the information you need, please feel free to contact us by e-mail or phone.

Overview

The StreamStor Serial FPDP (SFPDP) daughterboard is a mezzanine IO board that can be used with some StreamStor controllers such as the Amazon. Adding this daughterboard provides a high-speed data interface that conforms to the Serial FPDP specification (ANSI/VITA 17.1-2003). See www.vita.com for more information. SFPDP is a high-speed low latency data streaming serial communications protocol for use in high speed real time data transfer applications. Serial FPDP is directly related to the standard Front Panel Data Port (FPDP), deriving its serial protocol from the defined protocol and control signals of FPDP. See www.fpdp.com for more information. The StreamStor SFPDP is implemented using up to four optical transceivers over fiber optic cables. The board provides up to four independent SFPDP interfaces, each with sustained data rates up to 245 Mbytes/sec.

In reading the following sections on using this feature, it is important to be familiar with the American National Standard for Front Panel Data Port Specifications (ANSI/VITA 17-1998) and the Serial FPDP Specification (ANSI/VITA 17.1-2003). This manual is intended to clarify operation as it relates to the standard, not to educate one on the standard itself. For additional information about the standard, other FPDP products and manufacturers, and other technical details regarding FPDP, please visit www.fpdp.com and www.vita.com.

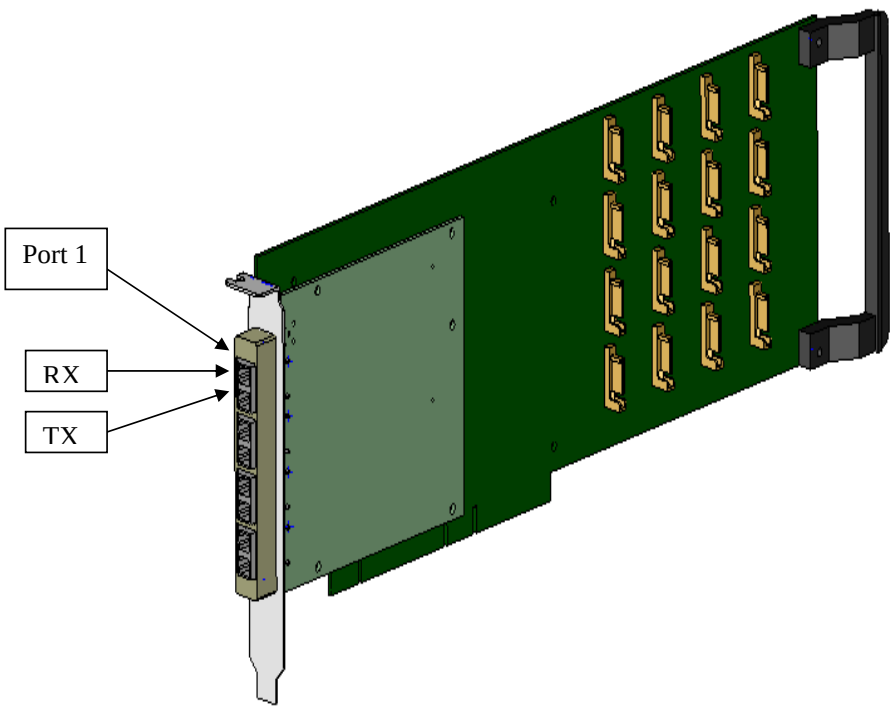
The StreamStor SFPDP interface is designed to meet and exceed the basic capabilities of SFPDP as defined in the SFPDP ANSI standard. The following sections describe:

- any optional SFPDP features StreamStor has implemented;
- any features that StreamStor has implemented as a superset to the standard;
- any known deviations from the ANSI standard;
- any clarifications that might otherwise be left open to interpretation; and
- the API functions necessary to configure an external port.

Optical Interface

The optical interface link speed supported by the StreamStor SFPDP is currently 2.5 Gbaud. The StreamStor SFPDP card supports up to four independent optical ports, each with sustained data rates in excess of 245 Mbytes/sec. Contact Conduant if you need support for optical link speeds of 1.0625 Gbaud or 2.125 Gbaud. One of the advantages of optical Serial FPDP is the extension of parallel FPDP over much greater distances, up to kilometers in some cases.

Each optical port consists of a transmit (TX) connection for transmitting data and a receive (RX) connection for receiving data. The RX connection is also used for flow control during data transmit. The RX connection is at the top of each port, with the TX connection directly beneath it. See picture below.

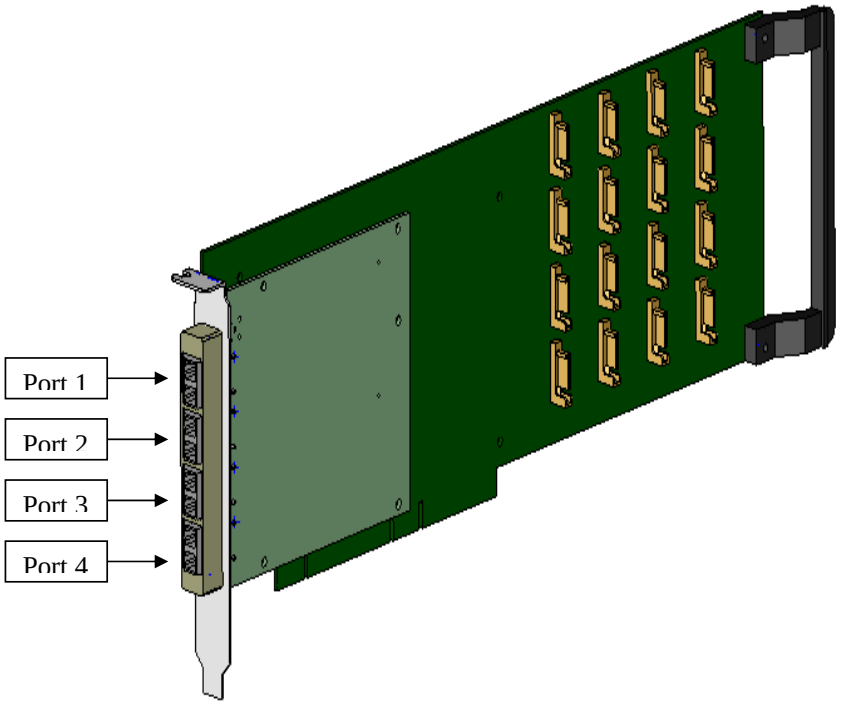


Deleted: <sp><sp><sp><sp><sp><sp>

Optical Port Description

The optical port (TX-RX pair) at the top of the SFPDP daughterboard is referred to as Port 1 or P1. The second optical port down from top is referred to as Port 2 or P2. The third down is referred to as Port 3 or P3, and the fourth down or the port closest to the bottom of the SFPDP board is referred to as Port 4 or P4. See picture below.

Deleted: <sp><sp><sp><sp><sp><sp><sp><sp><sp>



Programming

The SFPDP daughterboard is designed for flexibility so each optical port can be configured to run as either a receiver or transmitter, each with different options.

Configuring SFPDP with XLRSetMode

The StreamStor API function `XLRSetMode` is used to set the input/output path and functionality of the StreamStor. Table 1 lists the supported modes.

TABLE 1 – SFPDP I/O Modes

XLRSetMode Mode	Description
`SS_MODE_SINGLE_CHANNEL`	Receives and sends data over one channel
`SS_MODE_MULTI_CHANNEL`	Receives and sends data over multiple channels simultaneously. This mode is not valid if you for receiving or sending data over the PCI bus.
`SS_MODE_PASSTHRU`	Received data is not recorded. Instead, it is only passed out the bound output channel.
`SS_MODE_FORK`	Received data is recorded and simultaneously forked out the bound output channel.

Forking and PassThru

`XLRSetMode's SS_MODE_FORK` mode is used to record data to the StreamStor drives and simultaneously send that data out a single SFPDP physical port. In this mode, data can only be recorded to a single channel.

`XLRSetMode's SS_MODE_PASSTHRU` mode is used to receive data and simultaneously pass that data out a single SFPDP physical port. Data is not recorded to the StreamStor drives. In this mode, data can only be received from a single channel.

Configuring SFPDP with XLRSetDBMode

The StreamStor API function `XLRSetDBMode` is used to set the mode on the SFPDP daughterboard. The only valid mode for this daughterboard is `SS_SFPDPMODE_NORMAL`.

TABLE 2 – SFPDP Daughterboard Modes

XLRSetDBMode Mode	Description
`SS_SFPDPMODE_NORMAL`	Used to set both transmit and receive options.

`XLRSetDBMode` is also used to set daughterboard options. Table 3 lists the supported SFPDP daughterboard options. These options may be combined using a Boolean "or" operation before passing them into the `XLRSetDBMode` function.

TABLE 3 – SFPDP Daughterboard Options

XLRSetDBMode Option	Description
`SS_DBOPT_SFDPNRASSERT`	Receiver will assert "Not Ready" signal until recording starts. This option has no effect during playback of data.
`SS_DBOPT_SFPDP_CRC_ENABLE`	Enable SFPDP CRC generation during playback or CRC checking during recording.
`SS_DBOPT_SFPDP_CRC_DISABLE`	Disable SFPDP CRC generation during playback or CRC checking during recording.

The power-on default values for the SFPDP ports are "Assert Not Ready" is disabled and CRC generation is disabled. SFPDP Flow Control is always enabled for transmit and for receive. If this is an issue for your application, please contact Conduant Corp.

Setting up Channels

The StreamStor system can be configured in various ways to record or playback from the SFPDP ports or the PCI bus. The ports and the PCI bus are each mapped to a specific channel number, as shown in Table 4.

TABLE 4 – Channel Mapping

Channel Number	Channel Description
0	PCI
28	SFPDP Port 1 – Port at top of daughterboard.
29	SFPDP Port 2
30	SFPDP Port 3
31	SFPDP Port 4 – Bottom of daughterboard

The SFPDP daughterboard supports recording and playback of a single channel and over multiple channels.

Data can be recorded on a single channel, then that data can be played out any of the SFPDP daughterboard's physical ports. Note that if data was recorded in single channel mode, that data cannot be played out over multiple ports simultaneously.

In multi-channel mode, data can be recorded simultaneously over multiple channels then that data can be played out any of the SFPDP daughterboard's physical ports.

Input and output channels must be "selected" and "bound" before your application can record or playback data. The API function `XLRSelectChannel` is used to select a channel. When you select a channel, subsequent API calls (that are channel-specific) will be performed on that channel.

The API functions `XLRBindInputChannel` and `XLRBindOutputChannel` are used to identify which channel (or channels) will be used for data input and which channel (or channels) will be used for output.

The StreamStor SDK User's Guide describes each of the above API commands. The User's Guide also has a chapter called "Channel Description and Selection" that you may find useful.

Single Channel Operation

To configure the SFPDP daughterboard for recording or playback over a single channel, `XLRSetMode` can be called with one of the following modes.

- `SS_MODE_SINGLE_CHANNEL`
- `SS_MODE_PASSTHRU`
- `SS_MODE_FORK`

By default, the mode of the SFPDP daughterboard is `SS_MODE_SINGLE_CHANNEL.`

When in one of the above modes, the default binding for channel input and channel output is the PCI bus (channel 0). If you do not want to use the default binding, then you must call `XLRSelectChannel` and `XLRBindInputChannel` and/or `XLRBindOutputChannel`. If you do not want to use the default daughterboard options, you must also call `XLRSetDBMode.`

The following code fragment shows the sequence of API calls to use if you want to record over the PCI bus and then play data over SFPDP Port 1 (channel 28).

```
// Set up to record data over the PCI bus. Because by default
// the PCI bus is bound to the input channel, and because we
// are going to be recording on a single channel, we do not
// need to explicitly set the mode and bind the PCI bus.
// However, those commands are shown here for clarity.

XLRSetMode(xlrHandle, SS_MODE_SINGLE_CHANNEL);
XLRClearChannels (xlrHandle);
XLRSelect (xlrHandle, 0);
XLRBindInputChannel(xlrHandle, 0);
XLRecord(xlrHandle, 0, 1);
   … record for a while …
XLRStop(xlrHandle);

// Next, we must indicate the channel upon which the
// data was recorded and configure the output port.  To do
// this we must call XLRSelect with the channel number upon
// which the data was recorded, followed by the binding
// of the output channel.
XLRSelectChannel (xlrHandle, 0);

// Since we just selected channel 0, this call to bind
// the output will indicate to the StreamStor that
// channel 0 data is destined for output over channel 28.
XLRBindOutputChannel(xlrHandle, 28);
```

```
// In this example, we want to use CRC generation
// during playback, so we must call XLRSetDBMode to
// enable it on the output channel.
XLRSelectChannel (xlrHandle, 28);

// Since we just selected channel 28, this XLRSetDBMode
// command will be applied to channel 28.
XLRSetDBMode(xlrHandle, SS_SFPDP_MODE_NORMAL,
    SS_DBOPT_SFPDP_CRC_ENABLE);
XLRPlayback(xlrHandle, 0, 0);

   … play data for a while, then stop …
XLRStop(xlrHandle);
```

It is important to note the order of the API calls. The `XLRSelectChannel` command selects the channel that subsequent API commands will act upon. As shown in the example, we select the channel that was used when the data was recorded. Then, immediately after that, we bind the output channel. By calling the API commands in that order, it indicates to the StreamStor that it should retrieve the data that was recorded over channel 0 and then play that data out SFPDP physical port 1 (channel 28).

See the "C Coding Example" section of this document for a detailed example of how to record and playback in single channel mode.

Multi-Channel Operation

The following sections give a brief overview of how to record and playback in multi-channel mode. For more details on multi-channel mode, please refer to the SDK User's Guide chapter called "Channel Description and Selection." Multi-channel coding examples are in the StreamStor SDK installation directory in the *examples* subdirectory.

Recording over Multiple Channels

To record data over multiple channels simultaneously, your application must call `XLRSetMode` with the mode set to `SS_MODE_MULTI_CHANNEL`. (Note that multiple channels cannot be sent or received over the PCI bus.)

Before you begin recording, you must select and bind each channel that will be used during the recording. The following code fragment shows the sequence of API calls to use if you wanted data recorded to channels 28 and channels 29 simultaneously and you wanted CRC checking to be used on both channels.

```
XLRSetMode(xlrHandle, SS_MODE_MULTI_CHANNEL);

// Set up channel 28.
XLRSelectChannel (xlrHandle, 28);
XLRBindInputChannel (xlrHandle, 28);

// Since we selected channel 28 in the last
// XLRSelectChannel call, this XLRSetDBMode command will
// be applied to channel 28.
```

```
XLRSetDBMode(xlrHandle, SS_SFPDP_MODE_NORMAL,
    SS_DBOPT_SFPDP_CRC_ENABLE);

//Set up channel 29.
XLRSelectChannel (xlrHandle, 29);
XLRBindInputChannel (xlrHandle, 29);

// Since we selected channel 29 in the last
// XLRSelectChannel call, this call to XLRSetDBMode will
// be applied to channel 29.
XLRSetDBMode(xlrHandle, SS_SFPDP_MODE_NORMAL,
   SS_DBOPT_SFPDP_CRC_ENABLE);

// Now that both input channels have been bound for input
// and configured with the desired daughterboard options,
// we begin recording on the two channels.
XLRRecord(xlrHandle, 0, 1);
      … record for a while …
// When finished recording, call XLRStop, which will
// stop recording on both channels.
XLRStop(xlrHandle);
```

Playing back over Multiple Channels

To playback data over multiple channels simultaneously, your application must call `XLRSetMode` with the mode set to `SS_MODE_MULTI_CHANNEL`. (Note that multiple channels cannot be sent or received over the PCI bus.)

Prior to playing back data over multiple channels, you must:

- identify the channel that was used to record the data that you want to play and
- identify the physical port over which the data will flow.

Some possible scenarios are:

- Data was recorded over channel 28 and 29. You want to read data from channel 28.
- Data was recorded over channel 28 and 29. You want to retrieve data from channel 28 and play it out port 3 and retrieve data from channel 29 and play it out over port 4, simultaneously.

Note that data that is recorded on a single channel can only be read or played over a single physical port. For example, you cannot record data on channel 28 and then play that data out ports 2 and 3 simultaneously.

Multi-channel coding examples are in the StreamStor SDK installation directory in the *examples* subdirectory.

Setting Options with XLRSetOption

The `XLRSetOption` function is used to set other miscellaneous options. Table 5 lists the options available on the SFPDP daughterboard.

TABLE 5 – Options	
XLRSetOption Option	**Description**
SS_OPT_FSMAPPED	Enables read while recording. See Application Note AN105 on the Conduant website for details on this feature.
SS_OPT_PLAYARM	Arms two-stage playback.

Serial FPDP Daughterboard LED Descriptions

The Serial FPDP daughterboard has ten LEDs labeled D1 through D10 on the upper edge of the board. Each LED has three possible states, Off, On, or Flashing. Table 6 lists what each LED indicates.

TABLE 6 – LED Descriptions		
LED	**Color**	**Description**
D1	Green	-On when SFPDP is up and functioning (Link Up) on laser 1. -Flashing when record or play enabled and Link Up.
D2	Green	-On when SFPDP is up and functioning (Link Up) on laser 2. -Flashing when record or play enabled and Link Up.
D3	Green	-On when SFPDP is up and functioning (Link Up) on laser 3. -Flashing when record or play enabled and Link Up.
D4	Green	-On when SFPDP is up and functioning (Link Up) on laser 4. -Flashing when record or play enabled and Link Up.
D5	Green	-On when board initialized and locked to 40 MHz system clock.
D6	Red	-On when active flow control suspend or Not Ready on laser 1. -Flashing when record or play enabled with no signal detected.
D7	Red	-On when active flow control suspend or Not Ready on laser 2. -Flashing when record or play enabled with no signal detected.
D8	Red	-On when active flow control suspend or Not Ready on laser 3. -Flashing when record or play enabled with no signal detected.
D9	Red	-On when active flow control suspend or Not Ready on laser 4. -Flashing when record or play enabled with no signal detected.
D10	Red	-On when link to Amazon is not sync'd/ has active flow control.

C Coding Example

The following C code shows how to set up the Serial FPDP Port 1, channel 28, as an input channel to record, and then Playback the data back through the Serial FPDP Port 2, channel 29. For simplicity, error handling is not shown. In fully functional code `xlrStatus` should always be checked to see if an error was returned from a StreamStor API function call.

```
int main(int argc, char *argv[])
{
   SSHANDLE             hTarget;
   XLR_RETURN_CODE      xlrStatus;

   xlrStatus = XLROpen(1, &hTarget);
     -
   xlrStatus = XLRSetMode(hTarget, SS_MODE_SINGLE_CHANNEL);
     -
   // Call XLRClearChannels() before setting up channels for recording
   xlrStatus = XLRClearChannels(hTarget);
     -
   // Call XLRSelectChannel() before calling XLRBindInputChannel()
   xlrStatus = XLRSelectChannel(hTarget, 28);
     -
   // Bind Serial FPDP Port 1(channel 28), the top SFPDP port, as the
   // input channel.
   xlrStatus = XLRBindInputChannel(hTarget, 28);
     -
   // XLRSetDBMode will set the mode and options on the
   // currently selected channel.  We selected channel 28
   // above, so with this call to XLRSetDBMode, we are
   // enabling CRC generation on channel 28.
   xlrStatus = XLRSetDBMode(hTarget, SS_SFPDPMODE_NORMAL,
       SS_DBOPT_SFPDP_CRC_ENABLE);
     -
   // Start recording.
   xlrStatus = XLRRecord(hTarget, FALSE, TRUE);
     -
   // Sleep a while, letting some data record.
   Sleep(5000);

   // Stop the recording.
   XLRStop(hTarget);

   // Set up to read the data we just recorded.
   xlrStatus = XLRSetMode(hTarget, SS_MODE_SINGLE_CHANNEL);
     -
   xlrStatus = XLRClearChannels(hTarget);
     -
```

```
   // Call XLRSelectChannel() before calling XLRBindOutputChannel().
   // Since we recorded on channel 28 above, we select channel 28
   // here in order to play that data back.
   xlrStatus = XLRSelectChannel(hTarget, 28);
   -
   // Bind the Serial FPDP Port 2 (channel 29), the second from the top
   // SFPDP port, as the output channel.
   xlrStatus = XLRBindOutputChannel(hTarget, 29);
     -
   // We just selected channel 29, so this call to XLRSetDBMode
   // will enable CRC generation on channel 29.
   xlrStatus = XLRSetDBMode(hTarget, SS_SFPDPMODE_NORMAL,
      SS_DBOPT_SFPDP_CRC_ENABLE);
     -
   //Playback the data recorded on Channel 28 through Port 2 (channel 29)
   xlrStatus = XLRPlayback(hTarget, 0, 0);
     -
   // Sleep a while, letting some data playback.
   Sleep(5000);

   // Stop the playback.
   XLRStop(hTarget);

   XLRClose(hTarget);
   exit(0);
}
```

Technical Support

Conduant wants to be sure that your StreamStor system works correctly and stays working correctly. In the event, however, that you are unable to get your system to work properly, or if a working system ceases to function, we will do all that we can to get your system back online.

Solving the problem is largely a matter of data collection and steps that must be taken one at a time. In order for us to better serve you, we ask that you take the time to perform the following steps prior to calling us. This way, you can provide us with the most meaningful information possible that will help us solve the problem.

Is the problem one that obviously requires replacement parts due to physical damage to the system? If yes, then please gather the information described below and report the problem to tech support, by phone or through the Conduant web site.

Have you confirmed that no cabling has been inadvertently disconnected or damaged while working around the equipment?

Is the StreamStor card properly seated in the PCI (CPCI/PXI) slot?

Do all the systems have good power connections and voltages?

Does the confidence test sscfg.exe (on Windows) or ssopen/sstest (on Linux) run OK?

Has the software installation been corrupted? Try re-installing software.

Have you checked the Conduant web site for technical bulletins?

Have you recently installed a new Linux kernel or compiler or a new Windows Service Pack?

If the above steps did not resolve the problem, then please initiate a trouble ticket on the support section of the Conduant website at www.conduant.com. Please provide as much information about your system and the problem as possible. We will do all that we can to resolve the problem as quickly as possible.